SPARSITY MAY CRY: 😭
LET US FAIL (CURRENT) SPARSE NEURAL NETWORKS TOGETHER!

Anonymous authors

Efficient Learning Lab@POSTECH
Junwon Seo

Motivation

- Sparse Neural Networks(SNN) are good!
  - Efficiency, adversarial robustness, out-of-distribution generalization, etc.

- Conventionally..
  - We evaluate SNN targeting ________?

Motivation

- Sparse Neural Networks(SNN) are good!
  - Efficiency, adversarial robustness, out-of-distribution generalization, etc.

- Conventionally..
  - We evaluate SNN targeting a single or a few tasks. (usually image classification)
    - Mnist, CIFAR-10/100, ImageNet, GLUE

Motivation

Table 7: Summary of Evaluation Tasks and Datasets Used in 100 Recent SNN Papers.

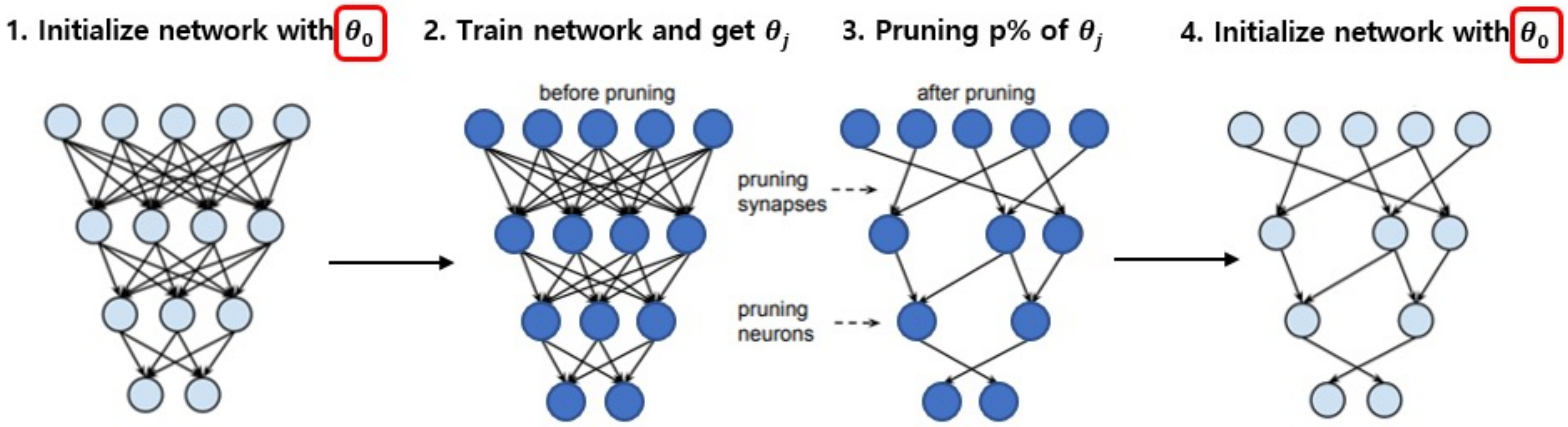
Task	Total #Paper	Datasets	#Paper
Image Classification	82	ImageNet	62
		CIFAR-10	59
		CIFAR-100	37
		MNIST	26
		Fashion MNIST	10
		SVHN	4
		Birds-200	1
		Flowers-102	1
		EMNIST	1
NLP Task	16	GLUE	9
		SQUAD	4
		WikiText-103	3
		WMT	5
		IMDB	1
		AAN	1
		LO	1
		OpenWeb text	1
		One Billion Word Benchmark	1
Face Recognition	3	LFW	3
		Youtube Faces	2
		CASIA-WebFace	1
Object Detection	3	COCO dataset	2
		PASCAL-VOL-2007	1
Speech Recognition	2	Google-12	1
		TIMIT	1
High-Resolution Reconstruction	2	Set5	2
		Set14	2
		B100	2
		Urban100	2
		Manga109	2
Image Generation	2	CIFAR-10	2
		ImageNet	1
		STL-10	1
Human Activity Recognition	1	HAR-2	1
Microarray Classification	1	Leukemia	1
		CLL-SUB-111	1
		SMK-CAN-18	1
		GLI-85	1
Hand Gesture Reconstruction	1	nvGesture	1
Regression Task	1	NYU Depth	1
3D Object Part Segmentation	1	ShapeNet	1
RL task	1	CartPole	1
		Acrobot	1
		MountainCar	1
		Atari suite	1
Vedio Deblurring	1	DVD	1
		GOPRO	1
		Real Blurry Videos	1
Vocabulary Speech Recognition	1	VS	1
		SWB	1

Motivation

~~ImageNet~~:
Too simple to evaluate

Pruning Method

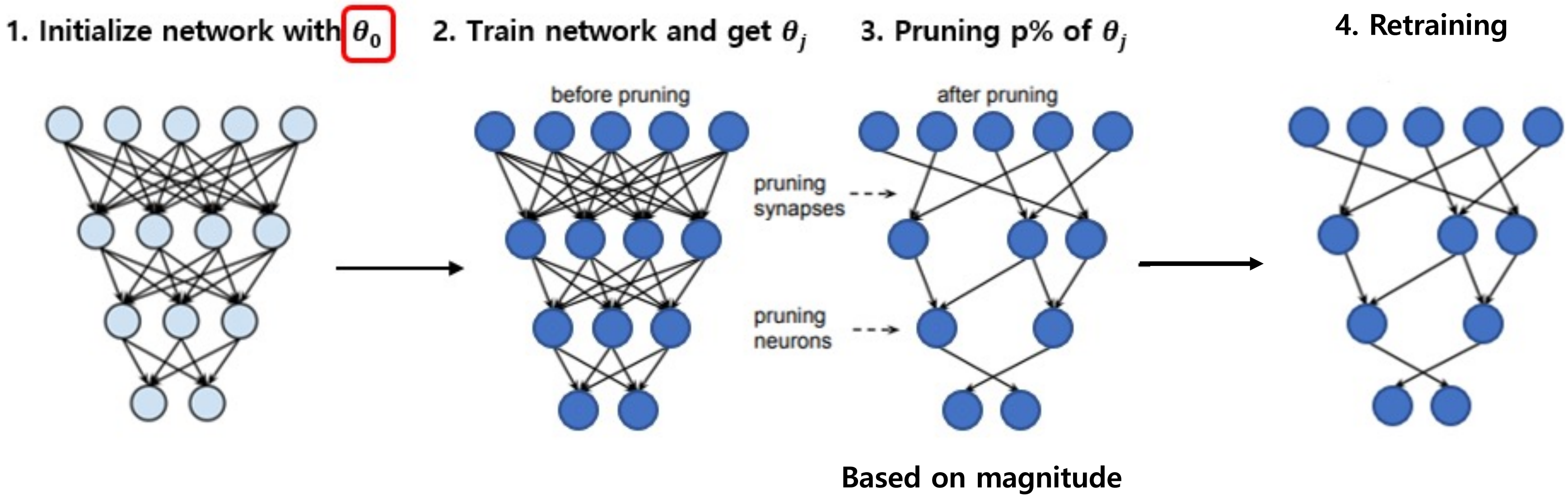
- Lottery Ticket Hypothesis (LTH)
  - Post-Training, Based on magnitude (Iterative adopt pruning)



Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks.

Pruning Method

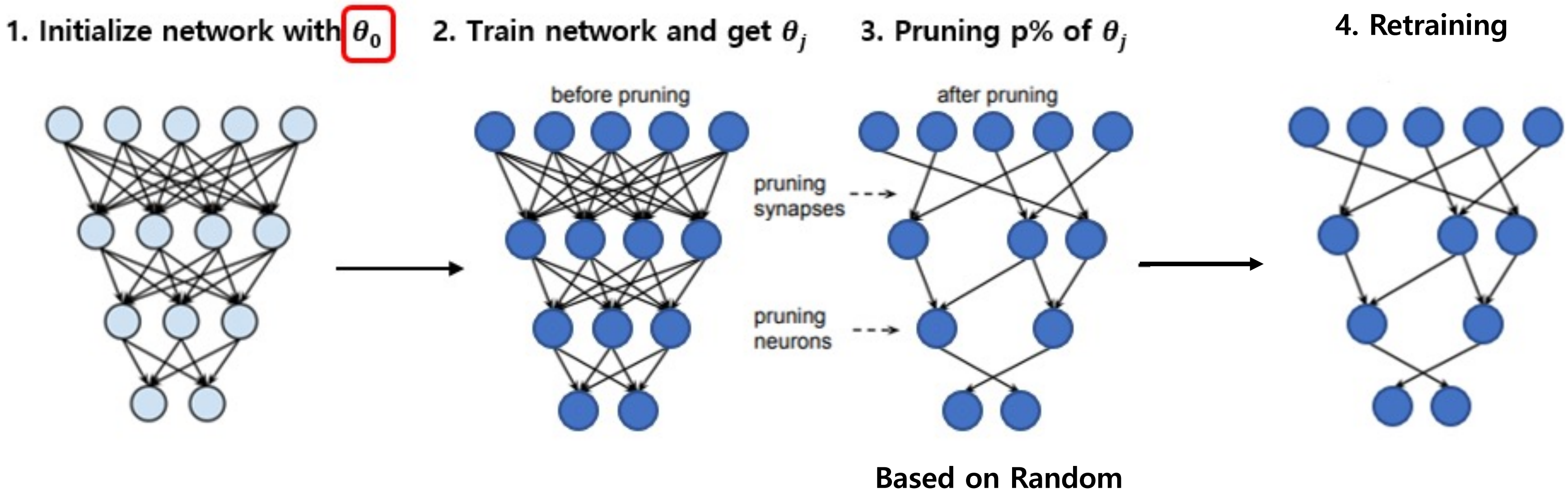
- Magnitude After Training (OMP (After))
    - Post-Training, Based on magnitude, one-shot



Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning.

Pruning Method

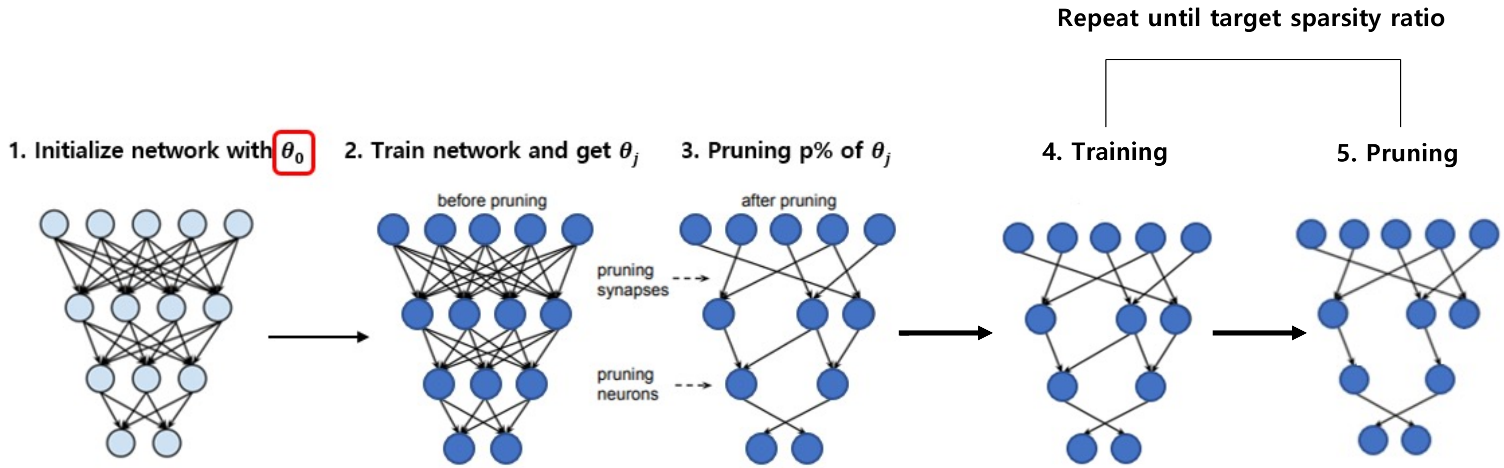
- Random After Training (Random (After))
  - Post-Training, Based on Random, one-shot



Studying the plasticity in deep convolutional neural networks using random pruning

Pruning Method

- Gradual Magnitude Pruning (GMP)
  - During-Training, Based on Magnitude

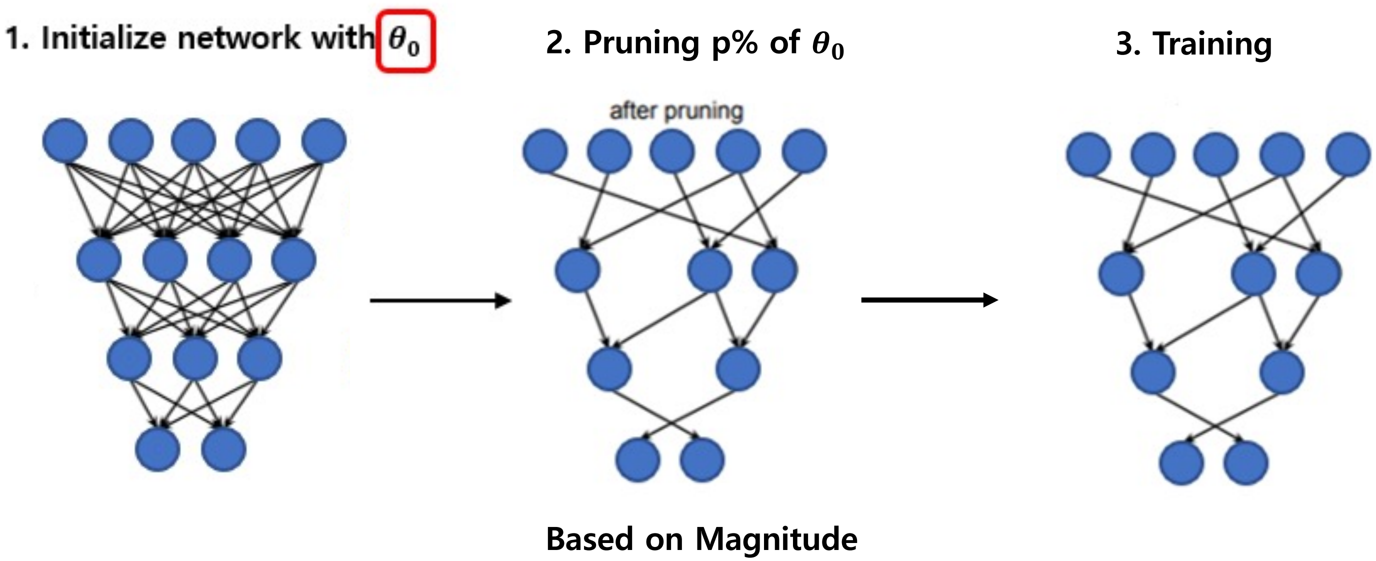


Based on Magnitude

Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression.

Pruning Method

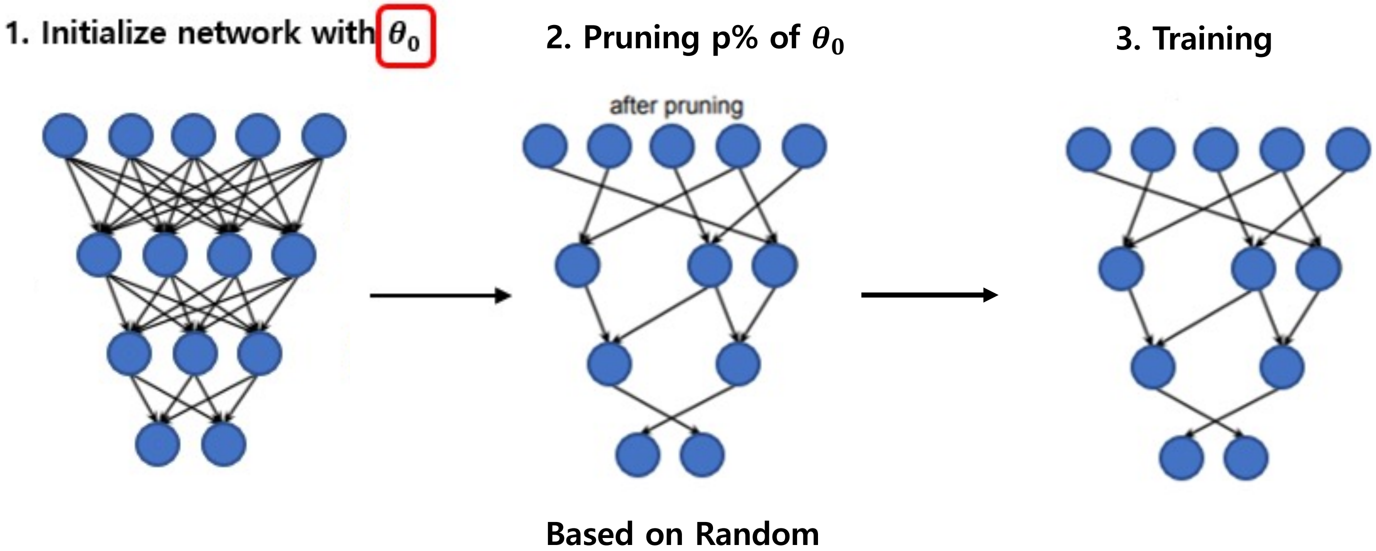
- Magnitude Before Training (OMP (Before))
  - Before-Training, Based on Magnitude



Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark?

Pruning Method

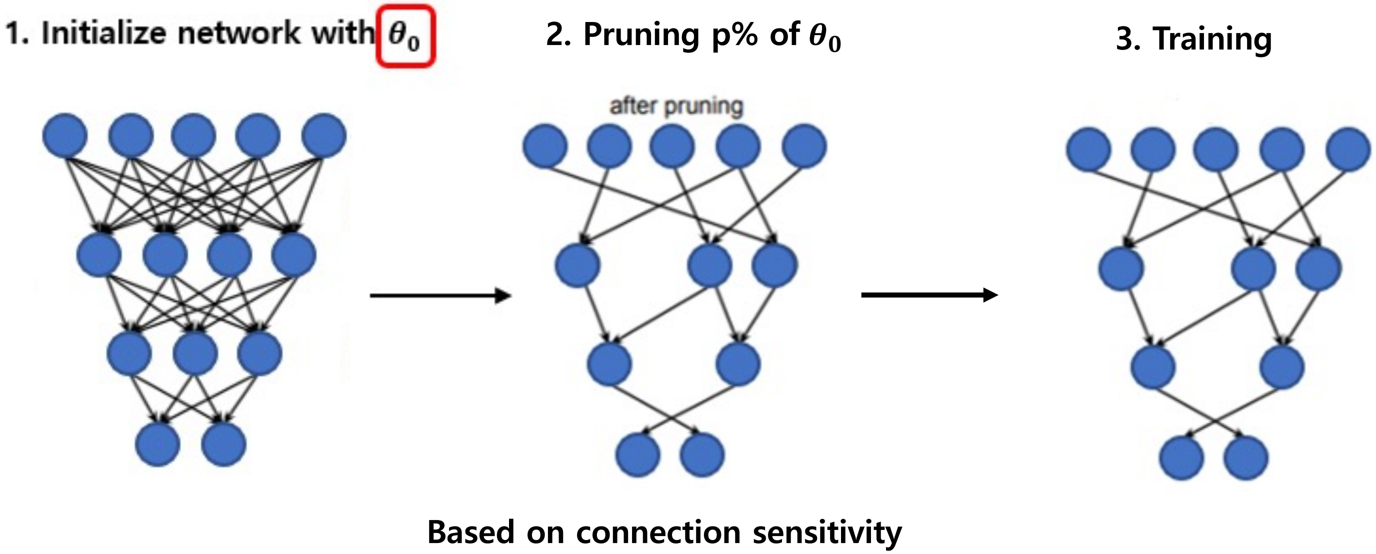
- Random Before Training (Random (Before))
  - Before-Training, Based on Random



Shiwei Liu, Tianlong Chen, Xiaohan Chen, Li Shen, Decebal Constantin Mocanu, Zhangyang Wang, and Mykola Pechenizkiy.
The unreasonable effectiveness of random pruning: Return of the most naive baseline for sparse training.

Pruning Method

- SNIP
  - Prior-Training, removes weight with the lowest connection sensitivity $|g \odot w|$



Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY.

Pruning Method

- Rigging the Lottery (RigL)
  - Update topology of SNN during training via prune-and-grow scheme.

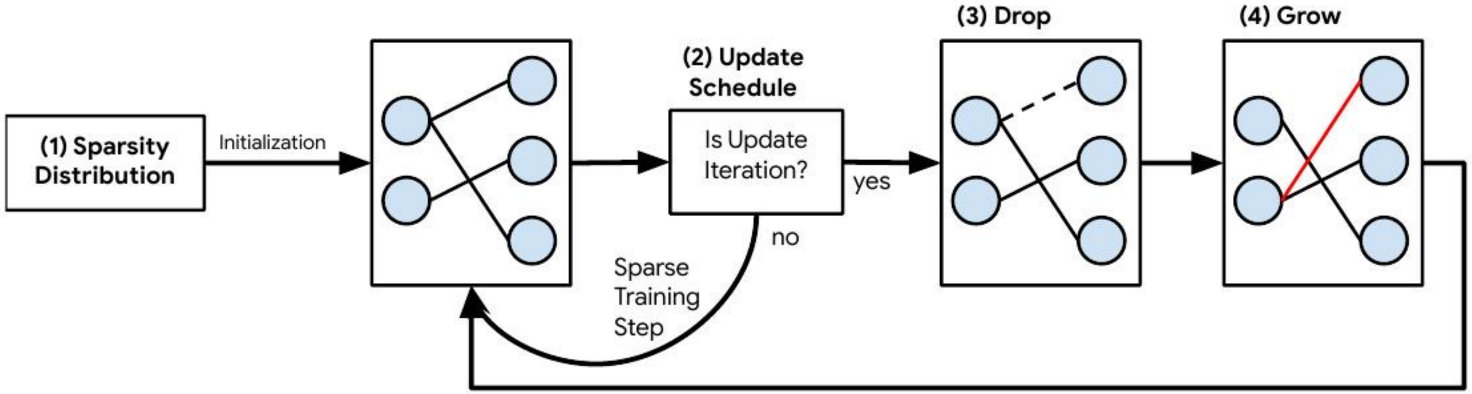


Figure 1: Dynamic sparse training changes connectivity during training to aid optimization.

Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners.

SMC-Bench

- Consists of 4 diverse and difficult tasks
  - Commonsense reasoning
    - Ask commonsense question about the world (RACE-M, RACE-H, WinoGrande, CSQA)
  - Arithmetic reasoning
    - Pose a question of a math problem and the model is asked to generate a mathematical equation (MAWPS, ASDiv-A, SVAMP)
  - Protein prediction
    - Ask a prediction of protein thermostability (HotProtein, Meltome Atlas)
  - Multilingual translation
    - Process multiple language using a single language model and the model perform translation across languages.
      (10 English-centric language pairs: Fr, Cs, De, Gu, My, Ro, Ru, Vi, Zh ↔ En)

Fr: French
Cs: Czech
De: German
Gu: Gujararti
My: Burmese
Ro: Romanian
Ru: Russian
Vi: Vietnamese
Zh: Chinese

Results – Commonsense reasoning

- Implementation Details

Models	RoBERTa	RoBERTa	RoBERTa
Dataset	CSQA	WinoGrande	RACE
Pre-trained Models	RoBERTa	RoBERTa	RoBERTa
Hidden Size	[1024]	[1024]	[1024]
FFN Inner Hidden Size	[4096]	[4096]	[4096]
Number of Layers	[24]	[24]	[24]
Learning Rate	[1e-5]	[1e-5]	[1e-5]
Weight Decay	[0.01]	[0.01]	[0.01]
Batch Size	[16]	[32]	[16]
Dropout	[0.1]	[0.1]	[0.1]
Attention Dropout	[0.1]	[0.1]	[0.1]
Clip Norm	[0.0]	[0.0]	[0.0]
Adam ϵ	[1e-06]	[1e-06]	[1e-06]
Adam β_1	[0.9]	[0.9]	[0.9]
Adam β_1	[0.98]	[0.98]	[0.98]
# Parameters	355M	355M	355M
Training Time	3000 steps	23750 steps	3 epochs
Wramup Time	150 steps	2375 steps	500 steps

- Test Accuracy: CSQA(77.3%), WinoGrande(76.3%), RACE-H(86.6%), RACE-M(81.3%)
- Human Accuracy: CSQA(89%), WinoGrande(94%), RACE(95%)

Results – Commonsense reasoning

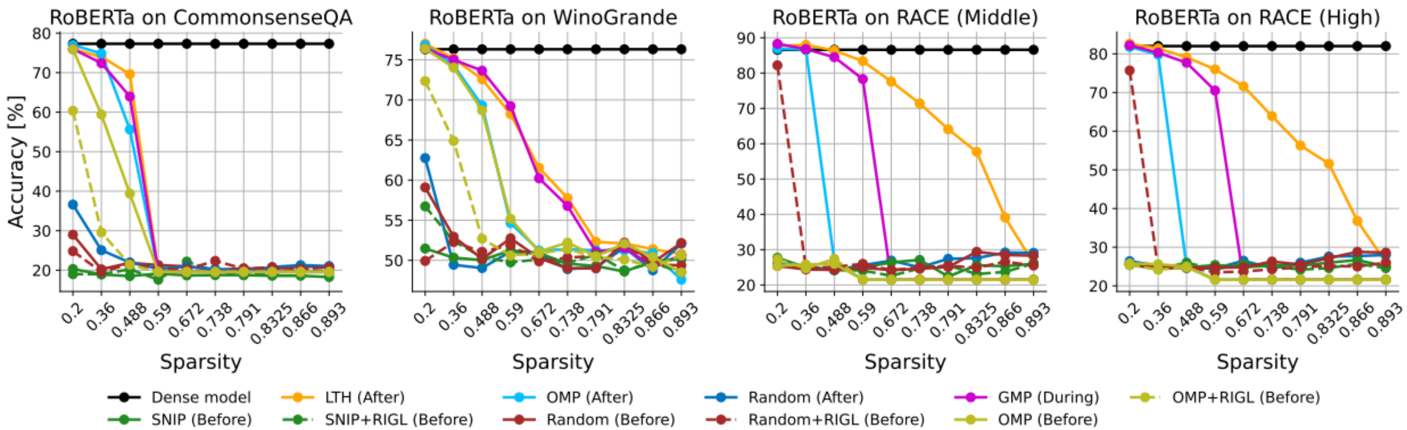
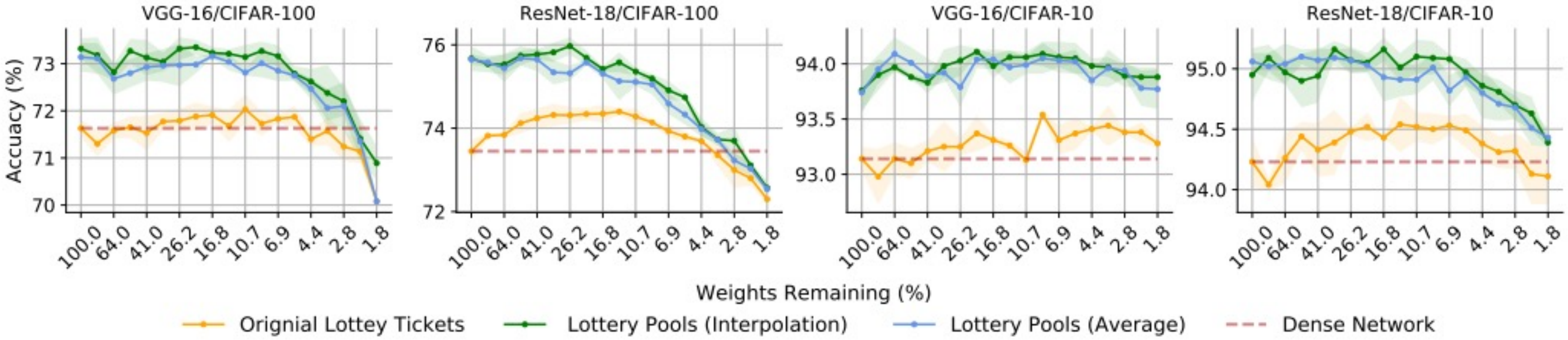


Figure 1: Commonsense reasoning performance of various sparse RoBERTa on CommonsenseQA, WinoGrande, and RACE.

1. All Sparse algorithm fail to find matching SNNs at trivial sparsities.

Results – Commonsense reasoning



(Contrast with the behavior of SNNs on the image classification task)

Results – Commonsense reasoning

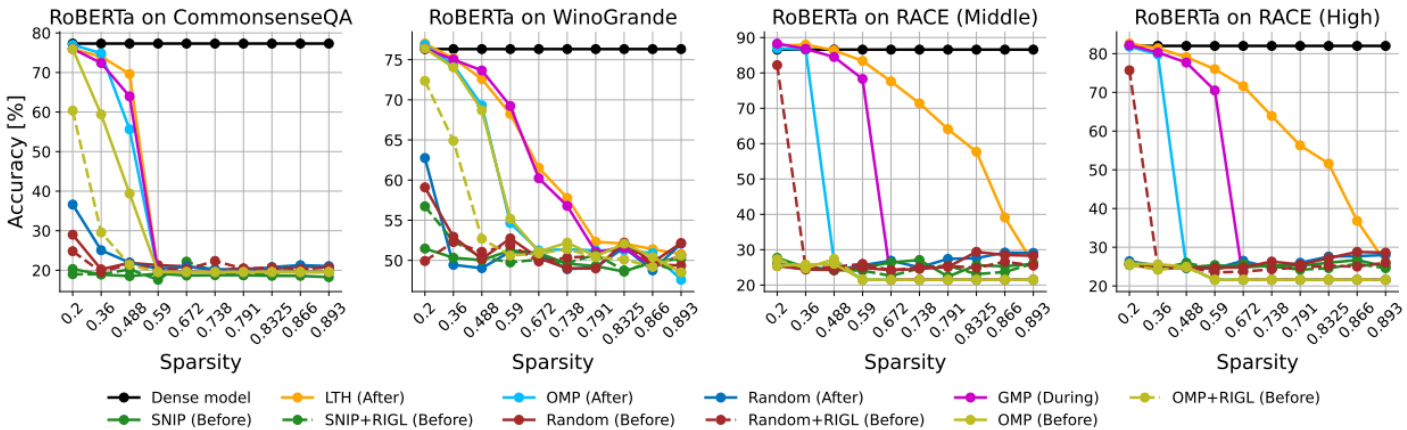


Figure 1: Commonsense reasoning performance of various sparse RoBERTa on CommonsenseQA, WinoGrande, and RACE.

2. The quality of SNNs on harder tasks suffers more from sparsity.

3. Post-training pruning consistently outperforms prior-training pruning.
(LTH, GMP, OMP(after) is good)

Results – Arithmetic Reasoning

- Implementation Details

Models	GTS	Graph2Tree
Dataset	MAVPS, ASDiv-A, SVAMP	MAVPS, ASDiv-A, SVAMP
Pre-trained Embedding	RoBERTa	RoBERTa
Embedding Size	[768]	[768]
Hidden Size	[512]	[384]
Number of Layers	[2]	[2]
Learning Rate	[1e-3]	[8e-4]
Weight Decay	[1e-5]	[1e-5]
Embedding LR	[8e-6]	[1e-5]
Batch Size	[4 (MAVPS, ASDiv-A), 8 (SVAMP)]	[4 (MAVPS, ASDiv-A), 8 (SVAMP)]
Dropout	[0.5]	[0.5]
Adam ϵ	[1e-08]	[1e-08]
Adam β_1	[0.9]	[0.9]
Adam β_1	[0.999]	[0.999]
# Parameters	140M	143M
Training Time	50 epochs	50 epochs

- GTS: LSTM (encoder), tree-based (decoder)
- Graph2Tree: graph transformer (encoder), tree structure (decoder)

Results – Arithmetic Reasoning

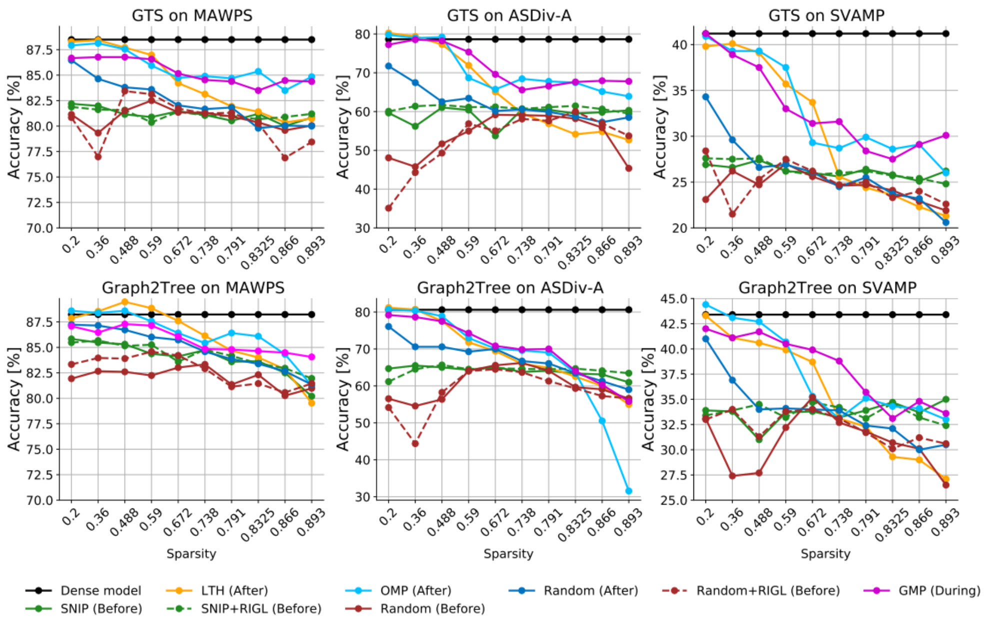


Figure 2: Arithmetic reasoning performance of various sparse GTS and Graph2Tree on MAWPS, ASDiv-A, and SVAMP.

- Overall accuracy trend is very similar to the commonsense reasoning
  → SNN can only match the dense performance when ratio is **lower than 48.8%**

Results – Arithmetic Reasoning

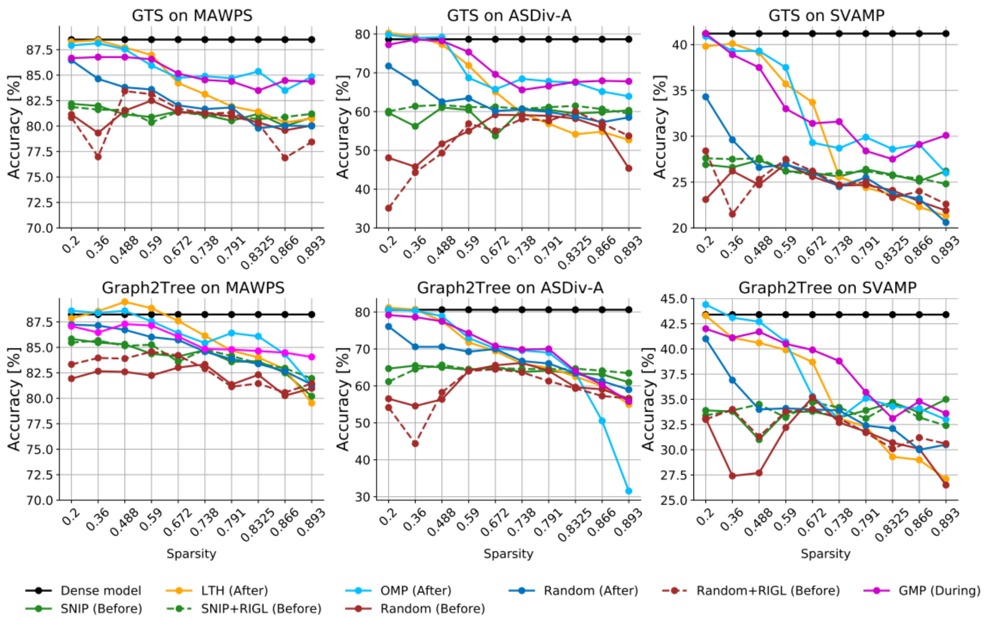


Figure 2: Arithmetic reasoning performance of various sparse GTS and Graph2Tree on MAWPS, ASDiv-A, and SVAMP.

- Overall accuracy trend is very similar to the commonsense reasoning
  → Difficulty makes SNNs sacrifice accuracy

Results – Arithmetic Reasoning

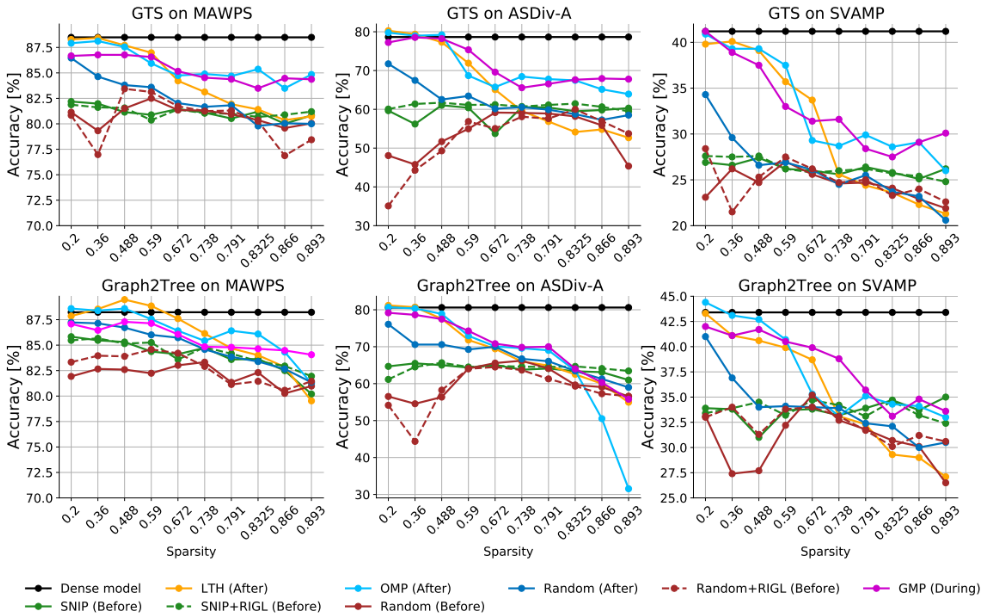


Figure 2: Arithmetic reasoning performance of various sparse GTS and Graph2Tree on MAWPS, ASDiv-A, and SVAMP.

- LTH method reaches lower accuracy than OMP and GMP at high sparsity levels.

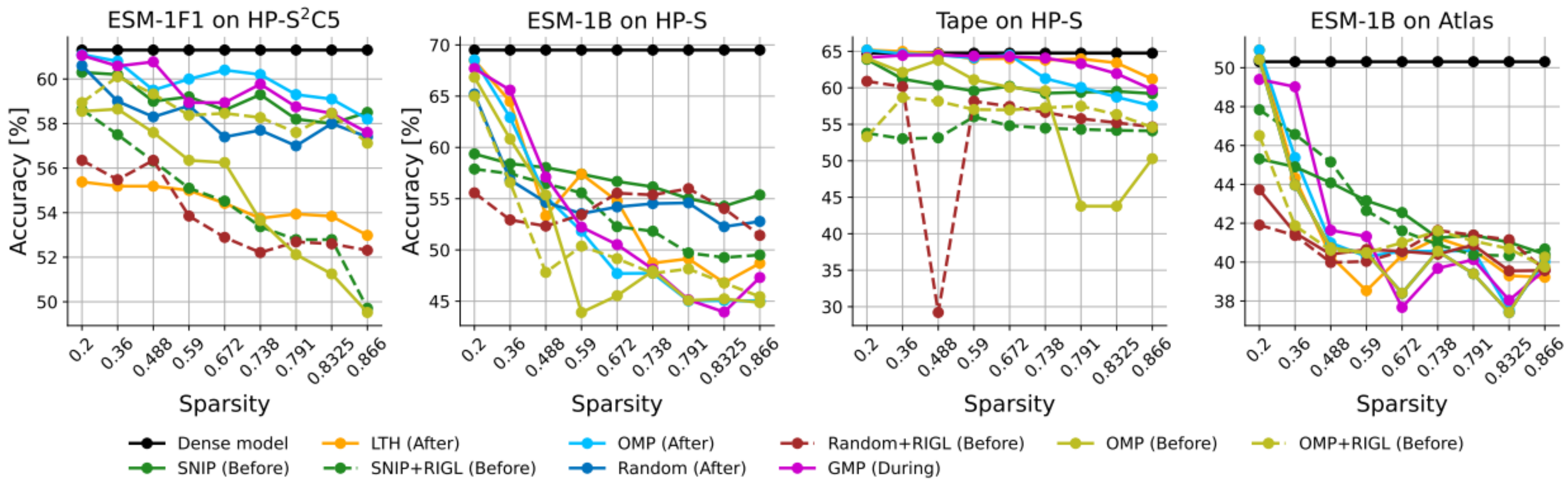
Results – Protein Thermal Stability Prediction

- Implementation Details

Models	TAPE	ESM-1B	ESM-IF1
Dataset	HP-S	HP-S^{2}C2, Meltome Atlas, HP-S	HP-S^{2}C5
Hidden Size	[768]	[1280]	[512]
Number of Layers	[12]	[33]	[20]
Learning Rate	[1e-4]	[2e-2 (head), 1e-6 (backbone)]	[2e-2 (head), 1e-4 (backbone)]
Weight Decay	[1e-2]	[1e-2]	[5e-2]
Batch Size	[16]	[?,2,3]	[4]
Dropout	[0.1]	[0.5]	[0.1]
Adam ϵ	[1e-06]	[1e-06]	[1e-06]
Adam β_1	[0.9]	[0.9]	[0.9]
Adam β_1	[0.999]	[0.999]	[0.999]
# Parameters	92M	650M	142M
Training Time	4 epochs	4 epochs	8 epochs

- All models use pretrained checkpoint.
- TAPE, ESM are based on Transformer.

Results – Protein Thermal Stability Prediction



- For ESM-1B, all SNN incur significant performance degradation whenever the sparsity level is larger than 20%
- For TAPE
  LTH, GMP, OMP(After) show satisfactory results before 59% sparsity.

Results – Multilingual Translation

- Implementation Details

Models	**mBART**	**mBART**	**mBART**
Dataset	2-to-2	5-to-5	10-to-10
Pre-trained Models	mBART	mBART	mBART
Hidden Size	[1024]	[1024]	[1024]
Number of Layers	[24]	[24]	[24]
Learning Rate	[3e-5]	[3e-5]	[3e-5]
Weight Decay	[0.0]	[0.0]	[0.0]
Batch Size	[16]	[32]	[16]
Dropout	[0.3]	[0.3]	[0.3]
Attention Dropout	[0.1]	[0.1]	[0.1]
Clip Norm	[0.0]	[0.0]	[0.0]
Adam ϵ	[1e-06]	[1e-06]	[1e-06]
Adam β_1	[0.9]	[0.9]	[0.9]
Adam β_1	[0.98]	[0.98]	[0.98]
# Parameters	680M	680M	680M
Training Time	40,000 steps	40,000 steps	40,000 steps
Wramup Time	2,500 steps	2,500 steps	2,500 steps

- Use 10 languages from the language pools for pretraining(Masked Language Modeling) and fine-tune 2, 5, 10 languages.

Results – Multilingual Translation

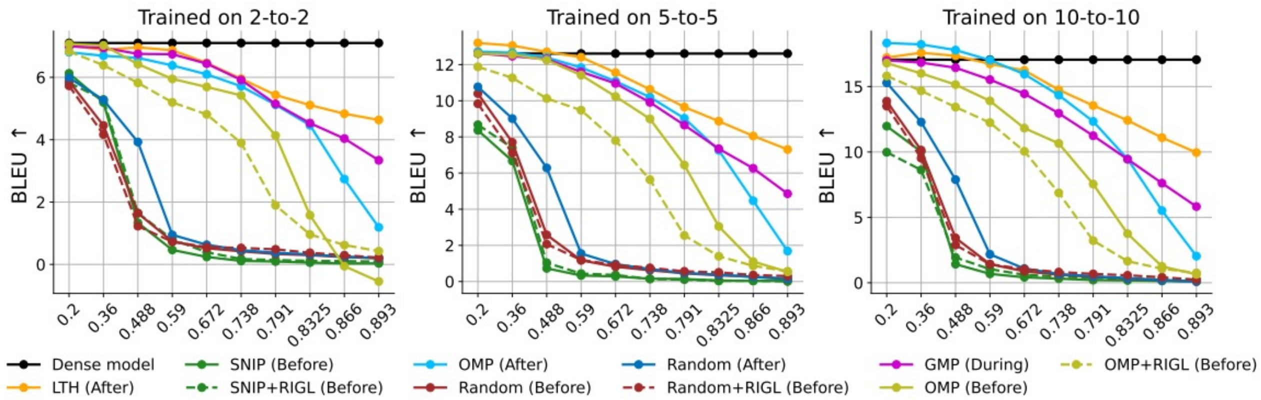


Figure 4: Multilingual performance of various sparse mBART. All models are tested on 10-to-10 multilingual translation and the averaged BLEU are reported.

- Fewer languages involved during fine-tuning leads to a more difficult translation for all languages. (means 2-to-2 is the most difficult)
- Similar to the previous experiment, models perform worse than the dense model. (besides OMP, LTH)

Results – Multilingual Translation

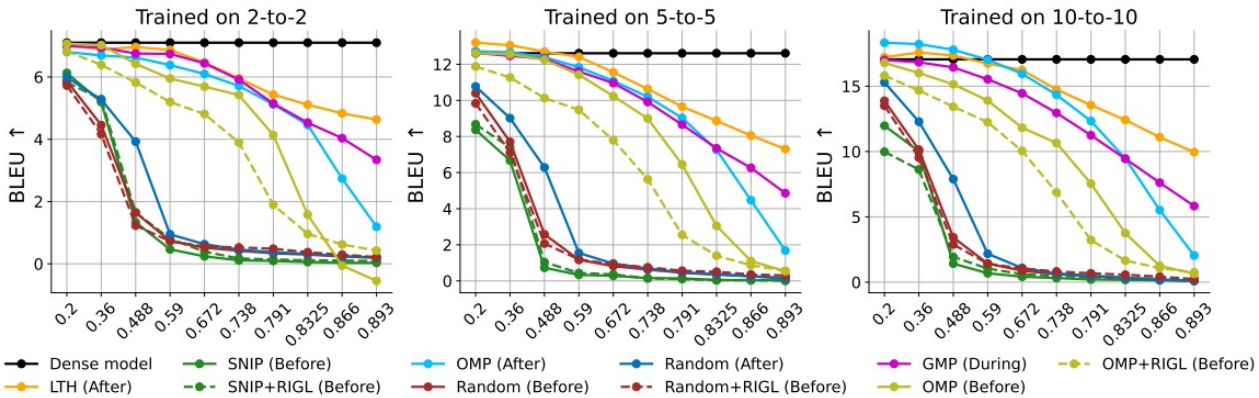
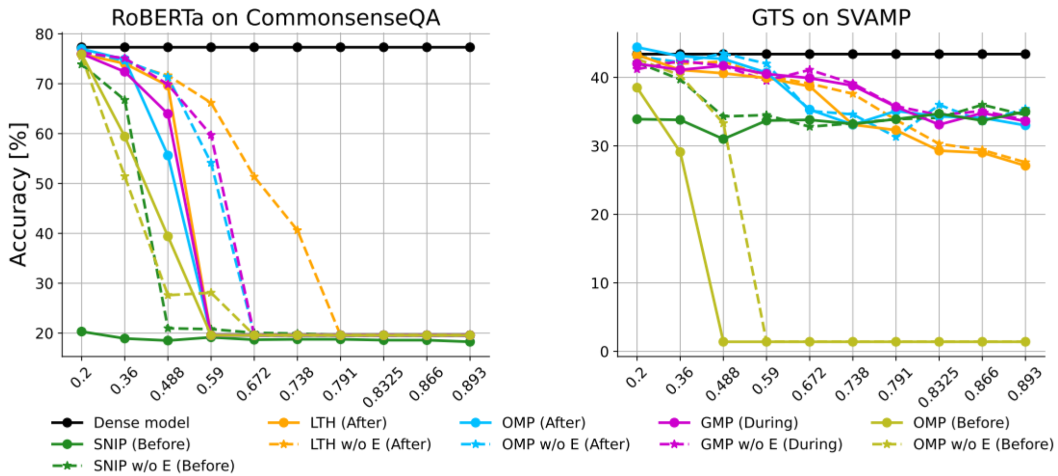


Figure 4: Multilingual performance of various sparse mBART. All models are tested on 10-to-10 multilingual translation and the averaged BLEU are reported.

- OMP, LTH also fail to match at 20%, 48.8%, 59%.
- Magnitude-based sparsifications (OMP, LTH, GMP) are “comparably” robust

The reason why SNNs Fail

- Pruning Embedding Layers or Not?
  - For dense model, pre-trained embedding play a crucial role. (21.4%)



  - Sparsification of embedding layers is not the root cause for the failure.

The reason why SNNs Fail

- Does layer collapse occur unexpectedly?
  - Do not observe severe layer collapse. (except SNIP (embedding layer))

  - Interesting thing is layerwise sparsities of different magnitude-based pruning approaches are **extremely similar.** (although performance gap exists)